[0173] One advantage of using the character information from the font texture file to define the dimensions of the polygon to receive the character texture is to minimize magnification or minification of the texels in the texture during rendering. As described with respect to **FIG. 4**D, magnification or minification may result in interpolations that degrade the quality of the pattern on the texture when rendered. In this embodiment, the initial size of the polygon used for the texture is selected to fit the character texture, which minimizes interpolation errors during rendering. Further, from their initial optimal sizes, the polygons for the characters may be stretched, shrunk or manipulated to some degree without too much degradation of the rendered text quality.

[0174] In this embodiment, if the polygons are scaled too much from their initial size rendered text quality may degrade. In this case, it may be desirable to use a font texture that is closer in size to the size of the font after scaling or to apply a technique such a MIP mapping. The gaming machine may be adapted to select a font texture of a particular size to match a desired font size and minimize scaling any errors. Thus, a font library of the present invention may include font textures for the same type of font at different sizes. With 3-D texture type fonts, the geometry information included with the font may be used for scaling and it may not be necessary to select a font texture of a particular size when scaling.

[0175] To reiterate, when using a 2D Texture type font, polygon geometry is not created to define the shape of a character, but instead 3D visible surface that a texture image can be applied to is created. The texture image may include the actual shape and look of the character. If a 3D Texture type font is used, then the font may contain the polygon information, which would define the 3D physical shape of the character (see **FIG. 8B** for example of a 3D texture type font). The texture image may be used to enhance it appearance. But, the polygon information may be used to define the character shape. Curve type fonts maybe treated the same as 3D Texture fonts except that the 3D physical shape of the character is defined by curves. Polygon may be created using the curve information and then a texture may be applied to the polygons

[0176] Another advantage of the 2-D texture method approach is that it reduces the number of polygons that need to be processed by the graphics software/hardware. In 3-D graphics systems, the ability to render scenes in real-time is a function of the number of polygons that need to be rendered. When the system has to process too many polygons, the performance of the system can be become degraded to the point where it is too slow to be of use in an operational environment. In the 2-D texture embodiment, the rendering of each character in a text string requires the processing of only two triangular polygons. Therefore, the method reduces the amount of polygons that need to be processed by the system as compared to an approach where a shape of each font in a character is represented by a large number of polygons.

[0177] There are numerous properties and features, which may be available through ActorText that may be used to provide text formatting and visual effects in the present invention. These additional features can also affect the generation of the character's 3D geometry. The following list describes some examples of features, which may be accessed by API function calls, scripts and models. However, the present invention is not limited to these examples. The commands are implemented to work in the context of the 3-D graphical rendering system used on the gaming machines or gaming devices of the present invention.

[0178] It is noted that commands described in the following paragraphs are high level commands. Each command may comprise a sequence of low-level commands or function calls that enable the high level commands to implemented in the 3-D graphical rendering system.

[0179] SetPosition may assign the x, y, and z positional coordinate for the location of a generated 3-D text object. SetScale may set the scaling value to be applied to the entire text string's size. SetRotation may be used to set the rotation values that may be applied to the entire text in the x, y and z-axis. The polygons defining a text character or text string may be manipulated like other 3-D objects defined in the 3-D gaming environment. SetPivotPoint may set the x, y and z positional coordinate for the location of the pivot point. The pivot point may be used as a reference location in the 3-D text object when rotating, scaling and positioning it. SetDisplayRegionSize may be used to set the text page's size (width, height and depth), which is used to contain the text string.

[0180] SetJustification may be used to set the type of justification used to position the text string in the text display region defined by the 3-D text page. There are several types of justification each can be combined together to form the desired justification effect. NONE no justification is applied to the text string. LEFT aligns the text string to the left side of the 3-D text page. RIGHT aligns the text string to the right side of the 3-D text page. HORIZONTAL CENTERED centers the text string horizontally in the 3-D text page. TOP aligns the text string to the top edge of the 3-D text page. BOTTOM aligns the text string to the bottom edge of the 3-D text page. VERTICAL CENTERED centers the text string vertically in the 3-D text page.

[0181] SetSizing may be used to set the sizing algorithm used on the text string. There are number of types of sizing algorithms that may be applied. NONE no sizing is applied to the text string. GROW TO FIT may size the text string to always fit inside the 3-D text page by shrinking or expanding the string's width and/or height. GROW TO FIT may keep the string's aspect ratio and operate on the string's width and height. GROW WIDTH TO FIT may change the string's width to always fit inside the 3-D text page's width by shrinking or expanding the string's width. GROW WIDTH TO FIT may change the string's aspect ratio and may operate on the string's width (height is not affected). GROW HEIGHT TO FIT may change the string's height to always fit inside the 3-D text page's by height shrinking or expanding the string's height. GROW HEIGHT TO FIT may change the string's aspect ratio and operates on the string's height (width is not affected).

[0182] SHRINK TO FIT may shrink the string to fit inside the 3-D text page when the string's width or height exceeds the boundaries of the 3-D text page. SHRINK TO FIT may keep the string's aspect ratio and may operate on the strings width and height. SHRINK WIDTH TO FIT may shrink the string's width to fit inside the 3-D text page's width when the string's width exceeds the 3-D text page's width.